

Methods, Data Structures, and Systems For Processing Media Data Streams

Arben Kryeziu

Priority

[0001] The present application claims priority to and is a continuation-in-part of co-pending U.S. Application entitled: "Methods, Data Structures, and Systems for Processing Media Data Streams," having serial no. 10/369,017, filed on February 19, 2003 the disclosure of which is incorporated by reference herein.

Technical Field

[0002] Embodiments of the present invention relate generally to media streaming, and more particularly to compressing, decompressing, and playing media data.

Background Information

[0003] Media streaming is commonplace in today's highly connected economy. Consumers use streaming technology to play live broadcast over the Internet or to play stored media data. Streaming presents a number of problems for media service providers and media content providers. One of these problems relates to managing limited network bandwidth and providing acceptable response times to consumers that desire the media data.

[0004] Response time can be improved with two techniques, upgrading hardware for both the media service providers and the consumers and improving streaming software technology to more rapidly transmit the media data from the service providers and play the media data for the

consumers on their computing devices. These two techniques are not mutually exclusive from one another. That is, often a combination of hardware and software techniques is used to improve delivery of media data over a network.

[0005] One software technique used in combination with streaming technology is data compression. Media data is voluminous and often includes video, image, graphical and audio data. Data compression attempts to represent that data in a smaller amount of physical space for purposes of transmission. When the compressed data is received from a consumer's computing device other software applications use decompression algorithms for reproducing the media data back to its original format for purposes of playing the decompressed media data within a media player. One commonly used compression format is the Motion Picture Expert's Group (MPEG) format. Often media players include a MPEG decompression application embedded within their instructions, such that decompression occurs seamlessly within the media player.

[0006] Although a variety of compression technologies are available in the industry, none of these compression technologies are customizable for purposes of improving the performance of streaming technology. In fact, the opposite is true in that the industry has been actively moving toward standard compression techniques, such as MPEG and others.

[0007] Industry standards for media data compression have benefits and drawbacks. Some benefits include the ability to freely develop applications to compress and decompress data using the standards and to integrate the compressed data in a variety of custom applications or of-the-shelf applications readily available in the marketplace. The drawbacks are that the compression used may not be modified easily and any changes made will take a long time to be integrated within the industry. Another drawback is that because a particular compression technique is readily known, it can lead to unauthorized consumption and distribution of media data to a large audience in a short period of time, where that audience often

lacks proper licenses to consume or distribute the media data.

[0008] Moreover, because an industry standard compression technique is rigid, there is little opportunity to alter attributes associated with the media data. Attributes can include such things as pixel resolution and sound quality. Attributes do not affect the media content but do affect the media quality. In some cases, a consumer would willingly forgo some quality if the media data could be received and played in a timelier manner. Furthermore, in some cases a consumer's computing environment may not even be equipped to handle higher pixel resolutions and sound quality, such that degradation in quality is not even appreciably noticeable by the consumer. If quality can be altered within the compression technique, then, in some instances, the size of the media data can be substantially reduced.

[0009] Conversely, quality need not be always lowered, since in some situations a consumer may desire and be capable of processing media data in a higher quality format, then what the media data is natively stored in. In these, situations the ability to improve quality within the compressed media data may be desirable. Additionally, from a media content provider's perspective, the ability to custom compress, and in some cases custom encrypt, media data provides added security benefits for their media content, since the media data is not easily acquired, distributed, and played when the format and play of the media data is in proprietary formats.

[0010] Therefore, there is a need for improved implementations and techniques for distributing media players, compressing media data, encrypting media data, and playing media data.

Brief Description of the Drawings

[0011] FIG. 1 is a flow diagram of a method for processing media data, in accordance with one embodiment of the invention.

[0012] FIG. 2 is a diagram depicting a media data server, in accordance

with one embodiment of the invention.

[0013] FIG. 3 is a diagram of a media data stream, in accordance with one embodiment of the invention.

Summary of the Invention

[0014] In various embodiments of the present invention, techniques for processing media data are presented. Media data is compressed and sent along with portions of a media player to a requestor. The pixel resolution for the compressed media data is determined based on a connection rate associated with the requestor. In some embodiments, the media data is also encrypted and the portions of the media player include the decryption algorithm necessary to decrypt the compressed media data.

[0015] More specifically and in one embodiment of the present invention, a method to process media data is described. A request for media data is received from a requestor. The media data is compressed with a custom pixel resolution based on a connection rate of the requestor. Next, portions of a media player and the compressed media data are streamed to the requestor at the connection rate.

Description of the Embodiments

[0016] Novel methods, data structures, and systems for processing media data are described. In the following detailed description of the embodiments, reference is made to the accompanying drawings, which form a part hereof, and in which is shown by way of illustration, but not limitation, specific embodiments of the invention that may be practiced. These embodiments are described in sufficient detail to enable one of ordinary skill in the art to understand and implement them, and it is to be understood that other embodiments may be utilized and that structural, logical, and electrical changes may be made without departing from the spirit and scope of the present disclosure. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the embodiments of the

inventions disclosed herein is defined only by the appended claims.

[0017] As used herein the phrase "media data" includes data that is related to multimedia such as, by way of example only, audio, video, graphical, image, text, and combinations of the same. Streaming refers to breaking media data up into configurable byte chunks, blocks, or frames and serially transmitting these pieces over a network to a one or more recipients' computing devices. The network can be hardwired (e.g., direct (point-to-point), indirect (e.g., Wide Area Network (WAN), such as the Internet), and others). The network can also be wireless (e.g., Infrared, Radio Frequency (RF), Satellite, Cellular, and others). Furthermore, the network can be a combination of hardwired and wireless networks interfaced together.

[0019] A media player is one or more software applications that are designed to receive, decompress, if necessary, decrypt, if necessary, and play media data in a requestor's computing environment. Media players can be any existing available media player designed and modified to automatically load, configure, and process media data according to the teachings of the present disclosure, or a custom-developed media player designed to achieve the same. In one embodiment of the present invention, the media player is a JAVA applet that can be downloaded, loaded, and executed within an application of a requestor's computing device (e.g., WWW browser, and others) without any manual intervention.

[0020] The media player of the present invention, is designed and performed according to the teachings of co-pending U.S. Application entitled: "Methods, Data Structures, and Systems for Processing Media Data Streams," having serial no. 10/369,017, filed on February 19, 2003 the disclosure of which is incorporated by reference herein. The media player is included within a media stream having at least a portion of desired media data, and it is designed to self-install and self-execute on a requestor's computing device for purposes of acquiring and playing the media data. In some embodiments, the media player is partially available from a World-Wide Web (WWW) browser page and exists as a partial script that when

activated is designed to configure and acquire the remaining portions of the media player necessary to play the media data. In these embodiments, the first portion of the media player (script) contacts and acquires the remaining portion of the media player from an initial stream of media data that includes some of the initial media data that a fully assembled and configured media player will consume.

[0021] The initial media data that is processed by various embodiments of the present invention can be in any existing media data format. Thus, this format can be Moving Picture Expert Group (MPEG) format, an Audio Video Interleaved (AVI) format, and a Quicktime Movie Format (MOV). The media data format can also be encrypted for purposes of validation or can itself be in a compressed format.

[0022] A requestor is a consuming user or automated application that desires to play media data. A requestor makes a request for media data from a media service provider. The requestor is associated with his/her own computing environment and connects with the media service provider over a network using an available connection rate of transmission.

[0023] FIG. 1 illustrates a flow diagram of a method 100 for processing media data, in accordance with one embodiment of the invention. Method 100 is implemented by one of more software applications on computer accessible media and is executed by a computing device (e.g., any device having processing and memory capabilities). In one embodiment, the method 100 is implemented as a media service provider on a server accessible over a network, such as the Internet.

[0024] Initially, a requestor makes a request to the processing of the method 100 for purposes of acquiring media data that the requestor desires to play. Accordingly, at 110, the request for the media data is received. Next, at 120, a connection rate associated with the requestor is determined. At 130, based on the connection rate a desired pixel resolution is determined and using that pixel resolution the media data is compressed. Finally, at 140, a custom media player and the compressed media data are

streamed to the requestor at the connection rate. In some embodiments, a first portion of the custom media player is already accessible to the requestor, such as via a WWW browser page, and that first portion assist the requestor in acquiring and configuring the remaining portions of the custom media player at 140.

[0025] In one embodiment, the connection rate associated with the requestor is determined and negotiated based on communications between the processing of the method 100 and the media player after it self-installs and self-executes on the requestor's computing device. That is, after the processing of the method 100 receives a request for the media data, the media player, or remaining portions of the media player as the case may be, is streamed directly to the requestor where it self installs and self executes, and immediately begins a direct dialogue of communication with the processing of the method 100. Once the media player is executing within the requestor's computing environment, it can manually interact with the requestor through interface screens to acquire a preferred pixel resolution for the desired media data and/or the connection rate of the requestor.

[0026] Alternatively, the media player can make its own determination for a desired pixel resolution for the media data based on the requestor's connection rate to the processing of the method 100. One way to do this is to assign a threshold value for a particular connection rate, such that connection rates below the threshold use a defined pixel resolution and connection rates above or at the threshold use another defined pixel resolution. Once the media player determines a connection rate and/or desired pixel resolution this is immediately communicated back to the processing of the method 100. Based on this information, the native media data is custom compressed with a custom pixel resolution that is optimal for the streaming the media data.

[0027] An optimal pixel resolution is one that is appropriate for the requestor based on the connection rate and optionally the computing capabilities of the requestor's computing device. A lower connection rate

can indicate that the pixel resolution for the media data can be decreased. If the pixel resolution is decreased than the physical space needed to represent the media data in a compressed format can also be decreased. Thus, there is less data to distribute from the processing of the method 100 to the requestor's computing device over the network. Accordingly, by custom decreasing the pixel resolution of the media data the throughput of streaming can be improved. Conventionally, this has not been the case, since there has not been a capability to custom compress media data. Moreover, there has been no attempt to custom alter pixel resolution prior to media streaming.

[0028] The pixel resolution need not always be decreased. For example, if the media data is natively stored with a medium resolution or quality and the connection rate and/or computing capabilities of the requestor are high, then it may be that the media data is custom compressed to have a higher pixel resolution, such that the requestor experiences improved quality.

[0029] Moreover, although various embodiments of this invention discuss pixel resolution as been altered for purposes of compression, it is readily apparent that any adjustable attribute of the media data can be custom altered prior to compression for purposes of customizing and improving the media streaming and media consumption. For example, sound quality of audio can be increased or decreased. Additionally, the custom quality can be manually identified by a requestor. That is, in some instances a requestor may want to see a video stream of desired media data, but may not want to hear the audio. This can happen when the requestor is deaf or when the requestor simply wants to see the video only. In such cases, the ability to custom exclude the audio can result in a much smaller compressed size for the media data and improve the performance of the media streaming. In a like manner, there may be situations where audio is desired but video is not desired.

[0030] In another embodiment, at 134, the compressed media data

can also be custom encrypted. The encryption algorithm can be customized for a particular requestor. The decryption algorithm can be custom configured within the media player. This provides an added level of security for media content providers, because each delivery instance of the media data can be protected according to licensing restrictions associated with a particular requestor. The media data can not be decompressed and decrypted without the custom media player which comes with the media data, and the media player only knows the decryption algorithm necessary to decrypt the media data. Moreover, the media player communicates with the processing of the method 100 and can be used to enforce licensing restrictions for a particular requestor. Thus, the media player will be of little use to someone who surreptitiously acquires it and the media content provider is assured of protected its rights in the media data.

[0031] In other embodiments, the media data can be natively stored at a location that is directly accessible to the processing of the method 100 or can be remote from the processing of the method 100. In one embodiment, the media data is acquired in real time and dynamically by the processing of the method 100, such as when the media data is being captured from a live broadcast, as depicted at 132.

[0032] FIG. 2 is a diagram depicting a media data server 200, in accordance with one embodiment of the invention. The media data server 200 is accessible over a network 210 to one or more media data requestors 220 and to one or more media content providers (not shown in FIG. 2). The media data server 200 includes a variety of software applications and resources designed to achieve the processing described herein and below.

[0033] The media data server includes a data store 201, a generic instance of a media player 202, and a streaming application 203. The data store 201 houses media data. The media data is not stored in duplicative data formats. That is, only a single instance of the media data resides in the data store 201. This is significant because conventionally media data is stored in a variety of formats in order to service requestors that can only use

a specific format associated with a specific media player. With the teachings of this invention, since the media player 202 is sent , or partially sent as discussed above, with the media data operability is ensured and there is no need to store the media data in more than one format. In one embodiment, the media data is natively stored within the data store 201 in an intermediate compressed format.

[0034] The media player 202 is software instructions designed to self install and self execute on computing devices of requestors 220. The media player is a generic instance and can be modified based on each request and/or requestor 220 for the media data. Thus, configuration files can be tailored for each instance of the media player 202, where these configuration files direct specific instances of the media player 202 to perform custom processing on specific requestor's computing devices. In this way, a particular instance of the media player 202 can include its own unique encryption algorithm and licensing restrictions. This ensures that the media data is controlled according to the desires of media content providers.

[0035] Instances of the media player 202 are streamed to requestors 220 when the media data server 200 receives a request from the requestors for portions of the media data housed in the data store 201. In some embodiments, as discussed above, portions of a media player 202 are available to the requestor 220 as a script that when accessed permits the requestor 220 to acquire the remaining portions of the media player 202 from the media data server 200. Once the instances of the media player 202 self install and self execute on the requestors' computing devices, the instances of the media player initiate a streaming dialogue with the streaming application 203 of the media data server 200. During this dialogue, a requestor's connection rate to the media server 200 is determined and a desired pixel resolution for desired portions of the media data is determined. This pixel resolution is then used when compressing the desired portions of the media data prior to streaming.

[0036] As was presented above, the pixel resolution can be

decreased in which case the physical size of the compressed media data is decreased, or the pixel resolution can be increased in which case the quality of the compressed media data is improved for the corresponding requestor 220. Moreover, in some embodiments, other attributes can be custom compressed with the media data, such as sound quality.

[0037] In some embodiments, the media data is captured and stored in the data store 201 as it is being recorded from a live broadcast and it is simultaneously custom compressed and streamed to a requestor 220.

[0038] With embodiments of the media data server 200, streaming is improved and made more secure, because compression can be customized to improve throughput performance or to improve quality. Moreover, security is achieved through custom distribution of media players 202 which are designed to execute custom dialogues with the media data server 200, to execute custom decryption algorithms, and to enforce custom licensing restrictions.

[0039] FIG. 3 is a diagram of a media data stream 300, in accordance with one embodiment of the invention. The media data stream 300 is implemented in a computer-readable medium and is accessible over a network. The media data stream 300 is created by a media service, such as the media data server 200 of FIG. 2. The media data stream 300 is consumed by a requestor 320.

[0040] The media data stream 300 includes all or some portions of a media player 301, a compressed and encrypted media data 302, and optionally restriction data 303. The media player 301 is self installing and self executing from the media data stream 300. That is, once the media player 301 is completely received on a requestor's computing device, it self loads and executes on that computing device and begins a dialogue of communication with the media service 310.

[0041] The media player 301 is designed with a unique decryption algorithm and decompression algorithm for decrypting and to decompressing the compressed and encrypted media data 302. In some

embodiments, each instance of a media data stream 300 includes its own unique instance of all or portions of the media player 301, such that each request for media data made by a requestor 320 requires a unique media player 301 instance to properly consume the media data. This provides an added level of security for protecting against unauthorized use and distribution of the media data. In some embodiments, the media player 301 also uses restriction data 303 that defines licensing rights associated with the media data. In this way, the media player 301 can use the restriction data 303 to ensure that the media data is conforming to licensing restrictions promulgated by a media content provider.

[0042] The media player 301 also establishes communication with the media service 320 for purposes of determining a proper pixel resolution or sound quality for the media data being requested by a requestor 320. One technique for doing this is to establish a pixel resolution rate for various connection rates and setting the pixel resolution based on a determined connection rate. The connection rate is associated with communications between the requestor 320 and the media service 310. The connection rate can be affirmatively communicated by the media player 301 to the media service 310 once the media player 301 is executing on the computing device of the requestor 320.

[0043] Once a determined pixel resolution and/or sound quality are determined, the media service custom compresses the desired media data having the defined pixel resolution and/or sound quality. In some instances, where resolution or quality is decreased this means that the compressed and encrypted media data 302 will be decreased in size. In other instances, where the resolution or quality is increased this means that the requestor 320 will experience improved media data quality during play by the media player 301.

[0044] The media data stream 300 is continuously and dynamically being populated and assembled by the media service 310 and streamed to the requestor 320. That is, the media service 310 does not statically

produce a single media data stream 300 before streaming the same over a network to the requestor 320; rather, the media service 310 first customizes and streams the media player 301 and optionally the restriction data 303, and thereafter continuously and regularly streams portions of the compressed and encrypted media data 302 to the requestor 320 for consumption. In some embodiments, the media data that is compressed and encrypted is also acquired dynamically by the media service 310 as it is streaming the same to the requestor 320, such as when the media data is associated with a live broadcast.

[0045] It is to be understood that the above description is intended to be illustrative, and not restrictive. Many other embodiments will be apparent to those of skill in the art upon reviewing the above description. The scope of embodiments of the invention should, therefore, be determined with reference to the appended claims, along with the full scope of equivalents to which such claims are entitled.

[0046] It is emphasized that the Abstract is provided to comply with 37 C.F.R. §1.72(b) requiring an Abstract that will allow the reader to quickly ascertain the nature and gist of the technical disclosure. It is submitted with the understanding that it will not be used to interpret or limit the scope or meaning of the claims.

[0047] In the foregoing Description of the Embodiments, various features are grouped together in a single embodiment for the purpose of streamlining the disclosure. This method of disclosure is not to be interpreted as reflecting an intention that the claimed embodiments of the invention require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive subject matter lies in less than all features of a single disclosed embodiment. Thus the following claims are hereby incorporated into the Description of the Embodiments, with each claim standing on its own as a separate exemplary embodiment.